

# BAB 01

## Mengenai Macro Excel dan VBA

Microsoft Excel adalah aplikasi yang sangat powerful dan fleksibel serta memiliki kemampuan yang sangat hebat. Namun demikian, Excel tidak memberikan opsi dan tidak mengizinkan Anda untuk melakukan hal-hal tertentu melalui user interface yang ada. Oleh karena itu, diperlukan pemrograman Excel yang akan dijelaskan dalam buku ini.

Contohnya ketika Anda ingin memilih sel-sel tertentu berdasarkan kriteria tertentu pula. Misalnya memilih semua sel yang memiliki nilai antara 100 hingga 200, atau menurut kriteria lainnya. Untuk melakukan hal ini dibutuhkan teknik pemrograman macro di Excel.

### 1.1 Hal-Hal yang Harus Diketahui Seputar Macro Excel

Pada umumnya, ketika belajar mengenai pemrograman Excel, ada beberapa hal yang perlu Anda ketahui. Berikut ini penjelasan beberapa hal tersebut.

#### 1. Visual basic editor.

Ini adalah lingkungan pemrograman di mana pemrograman Excel dilakukan. Ini sering disebut sebagai Visual basic editor atau Excel VBA IDE (integrated development environment).

#### 2. Dasar-dasar pemrograman di VBA.

Berikutnya Anda perlu belajar dasar-dasar dari bahasa pemrograman yang digunakan oleh Excel. Bahasa ini disebut Visual basic for applications atau sering disingkat sebagai VBA.

Sebetulnya, VBA tidak cuma dipakai oleh Excel saja, tapi juga dipakai di semua aplikasi berbasis Microsoft Office, seperti Access, Word, PowerPoint, dan sebagainya.

Aplikasi-aplikasi yang menggunakan VBA ini sering disebut sebagai aplikasi host untuk VBA.

### 3. Model objek dan Model objek dari Excel.

Tiap aplikasi host VBA seperti Word, Excel dan PowerPoint selalu memberi pelengkap pada bahasa VBA dengan menyediakan model objek. Model objek ini adalah objek yang akan diatur dengan kode menggunakan objek yang khusus untuk aplikasi tersebut.

Contohnya VBA di Excel mengenal beberapa model objek, yang nantinya akan ditangani oleh VBA. Antara lain workbook, worksheet, sel, baris, kolom, range, chart, pivot tabel, dan sebagainya.

Programmer Excel harus familier dengan berbagai model objek. Dan model objek di Excel ini sangat banyak jumlahnya. Bahkan kalau dihitung, ada lebih dari 200 objek yang berbeda. Tapi jangan takut duluan, karena Anda harus familier paling tidak dengan beberapa objek saja untuk bisa memprogram di VBA Excel.

Dan untungnya, kebanyakan action yang Anda perlukan hanya berkaitan dengan kurang lebih tujuh objek saja, seperti Application, Range, WorksheetFunction, Workbook, Worksheet, PivotTable dan Chart.

Jadi, apabila Anda tertarik memprogram agar bisa mengefisienkan pekerjaan di Excel atau memang mau menjadi programmer profesional yang *code for food*, maka Anda akan sangat menikmati keuntungan kalau bisa memanipulasi Excel pada level program.

Dan karena pengaksesan objek model di Excel ini menggunakan bahasa visual basic for application, maka ketika Anda sudah menguasai VBA untuk Excel, maka Anda juga berarti setengah menguasai bahasa vba lain untuk Word, PowerPoint dan Access. Jadi, satu merengkuh dayung, dua tiga pulau terlampaui.

## 1.2 Apa Itu Bahasa Pemrograman

Membuat macro di Excel membutuhkan penguasaan tentang bahasa pemrograman. Apa sebenarnya bahasa pemrograman itu? Bahasa pemrograman atau dalam bahasa Inggris disebut *programming language* merupakan bahasa khusus dan terbatas yang hanya dipahami oleh komputer pada tingkatan tertentu.

Secara umum, bahasa pemrograman dibagi menjadi tiga tingkatan:

1. Bahasa pemrograman yang didesain untuk memanipulasi komputer pada level rendah, yaitu untuk memanipulasi sistem operasi atau bahkan pada level hardware. Ini disebut bahasa pemrograman tingkat rendah alias low level. Contoh bahasa pemrograman di level ini adalah bahasa assembly.
2. Bahasa pemrograman yang digunakan untuk membuat aplikasi standar, seperti program MS Office, AutoCAD, dan sebagainya. Ini adalah bahasa tingkat tinggi atau high-level language. Contohnya BASIC, COBOL, Fortran, C, C++ dan Visual Basic.
3. Bahasa pemrograman yang ditujukan untuk memanipulasi program aplikasi, disebut bahasa pemrograman level aplikasi. Contohnya VBA Excel ini.

Pembagian bahasa pemrograman ke tiga tingkatan di atas, tidaklah baku karena itu wajar kalau menurut beberapa orang, dan banyak pihak lain yang juga menggolongkan beberapa bahasa pemrograman tertentu ke level yang lain.

Tiap bahasa pemrograman memiliki sintaks-sintaks yang berbeda. Beberapa sintaks dari bahasa pemrograman lebih mudah dipahami, dan sebagian yang lain lebih sukar.

Ketika melakukan pemrograman, ada dua hal yang perlu Anda perhatikan agar bisa menghasilkan kode program yang bagus:

1. Dalam memprogram, pentingkan keterbacaan, dan bukan malah membuat program yang susah dibaca agar dianggap lebih mahir.
2. Isi program Anda dengan komentar-komentar yang banyak.

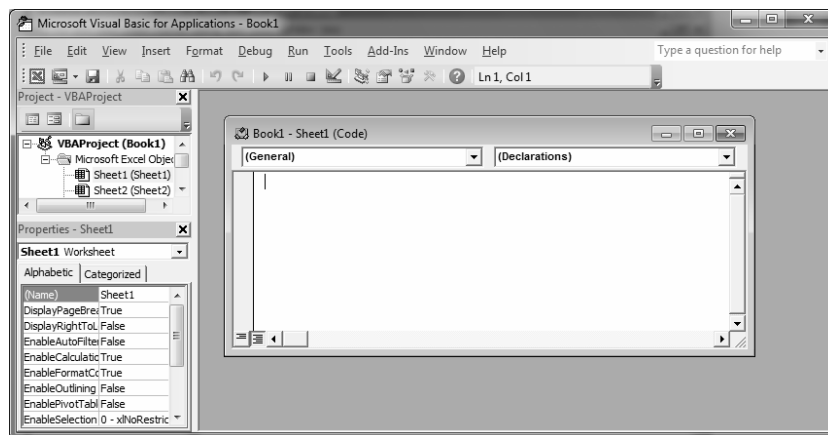
Jangan sering-sering menggunakan keyword yang membuat program susah dibaca, misalnya menggunakan statement GoTo yang memungkinkan program meloncat-loncat tidak karuan.

Program GoTo dirancang untuk me-redirect penjalanan program ke lokasi lain. Ini memang praktis, tapi membuat program tidak bisa dibaca dengan baik.

### 1.3 Mengetahui Environment Visual Basic Editor

Langkah pertama untuk menjadi programmer VBA di Excel adalah mengetahui lingkungan kerja IDE (integrated development environment) yang dipakai untuk melakukan pemrograman VBA.

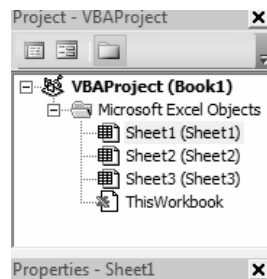
Cara untuk menjalankan visual basic editor ini adalah dengan klik pada tombol ALT+F11 di keyboard saat membuka Excel, maka terlihat tampilan berikut.



*Gambar 1.1 Tampilan dari Visual Basic Editor*

#### 1.3.1 Jendela Project

Jendela ini lokasinya paling kiri atas di bawah toolbar. Tampilannya terlihat seperti berikut.



**Gambar 1.2 Jendela Project**

Perhatikan bahwa project explorer ini memiliki bentuk bertingkat mirip dengan Windows Explorer. Node bagian paling atas merepresentasikan proyek VBA Excel yang sedang dibuka.

Tampilan dari tiap proyek bisa diperluas atau dipersempit dengan mengklik kotak kecil di sebelah kirinya mirip dengan Windows Explorer. Hanya ada satu proyek untuk satu workbook Excel.

Tiap proyek memiliki nama, di mana programmer bisa menggantinya kalau perlu. Nama default yang diberikan di proyek selalu seperti berikut:

NamaProjek (NamaWorkbook)

Di mana NamaProjek adalah nama dari proyek, sementara NamaWorkbook adalah nama dari workbook Excel.

Sebuah proyek berisi konten-konten tertentu. Anda bisa melihatnya di bagian bawah dari top level langsung. Di situ ada node-node yang bernama:

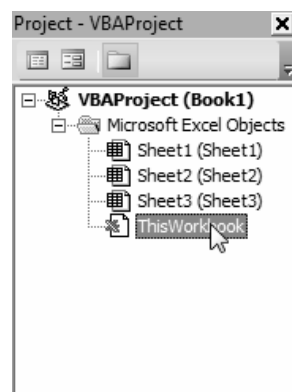
1. Microsoft Excel Object
2. Form
3. Modules
4. Class

Di bawah node Microsoft Excel Objects, Anda bisa melihat node-node untuk tiap worksheet dan chartsheet yang ada di workbook. Ada juga node khusus yang bernama ThisWorkbook yang merepresentasikan workbook itu sendiri. Kode ini menyediakan akses ke jendela kode untuk semua objek, di mana programmer nantinya bisa menuliskan kodenya.

Di bawah node Form, ada node untuk tiap form di proyek. Form sering pula disebut UserForm yang merupakan antarmuka form yang Anda buat di VBA Excel. Di bawah node Modules, ada node untuk tiap modul kode yang ada di proyek. Modul kode ini sering juga disebut modul standar. Sementara di bagian bawah modul Classes, terdapat node untuk tiap modul class di proyek.

Fungsi utama dari Project Explorer adalah agar kita bisa melihat apa saja yang ada di proyek kita dan memudahkan menavigasi komponen yang ada di proyek. Dengan mengklik worksheet atau node Forms, kita bisa memilih objek yang akan dilihat.

Di bawah tiap node di Project Explorer dengan nama Microsoft Excel Objects, selalu ada node yang bertuliskan ThisWorkbook. Node ini mengacu pada workbook dari proyek beserta modulnya yang bertugas menyimpan kode event untuk workbook.



**Gambar 1.3 Objek ThisWorkbook**

Jadi, tujuan event ini agar programmer VBA bisa menuliskan kode yang akan tereksekusi saat event tersebut dipicu. Contohnya:

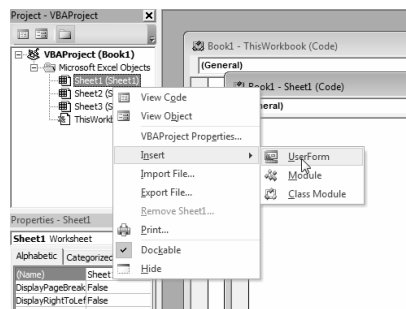
1. Event Open yang terjadi ketika workbook dibuka.
2. Event BeforeClose yang terjadi sebelum workbook ditutup.
3. Event NewSheet yang akan terjadi ketika worksheet baru ditambahkan ke workbook.
4. Event BeforePrint yang terjadi sebelum workbook atau apa pun tercetak.

Di bagian bahwa tiap objek Microsoft Excel Objects, terdapat node untuk tiap sheet. Tiap node ini merepresentasikan worksheet atau chartsheet dan juga komponen lainnya. Excel mengenal 7 event berkaitan dengan worksheet dan chartsheet.

Sebuah modul atau standard module, adalah modul kode yang berisi prosedur umum, baik fungsi dan subrutin. Prosedur ini bisa berupa makro yang didesain untuk dijalankan user. Atau bisa mendukung program yang digunakan oleh program lain.

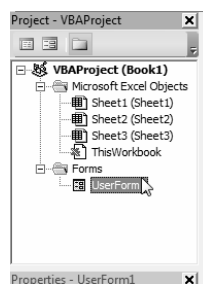
Modul class merupakan modul kode yang berisi kode berkaitan dengan objek kustom. Kode di sini nantinya bisa digunakan untuk yang lain.

Objek UserForm digunakan untuk membuat kotak dialog yang terkostumisasi. Anda bisa memasukkan objek-objek ini ke objek Microsoft Excel dengan klik kanan lalu klik **Insert** > jenis objek.



**Gambar 1.4 Menu Insert > jenis objek untuk memasukkan objek**

Kalau sudah, maka objek tersebut akan masuk.

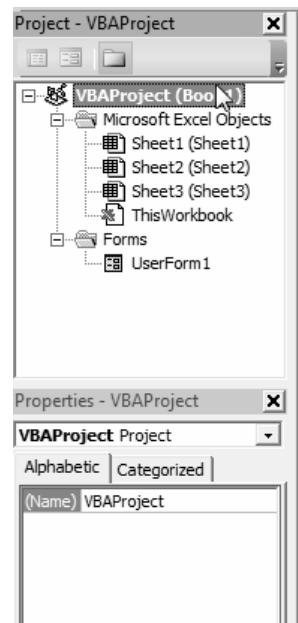


**Gambar 1.5 Objek Form yang dimasukkan sudah terlihat di jendela Project**

### 1.3.2 Jendela Properties

Jendela Properties akan menampilkan properti objek dan memungkinkan programmer mengubahnya. Tampilan dari jendela Properties ini berubah-ubah sesuai dengan item dari objek yang dipilih.

Ketika modul nama proyek dipilih, maka hanya terlihat Name saja.

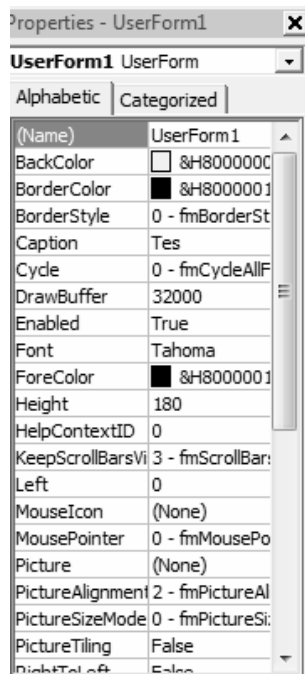


***Gambar 1.6 Ketika nama proyek dipilih, maka hanya atribut Name yang ditampilkan di Properties***

Jendela Properties ini bisa dipakai untuk mengubah beberapa properti dari objek ketika kode sedang tidak dijalankan, alias pada saat desain saja. Tapi beberapa properti ini hanya bersifat read only dan tidak bisa diubah.

Sementara kalau yang diklik objek lain seperti Form, maka tampilan jendela Properties akan terlihat berubah seperti berikut ini.

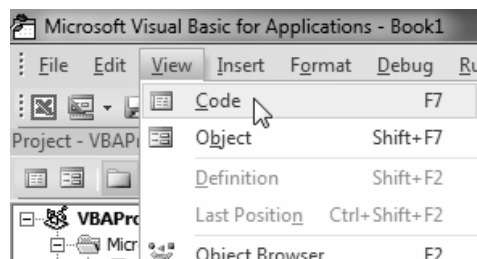




*Gambar 1.7 Tampilan Properties dari Form akan berkaitan dengan objek-objek dari Form ini*

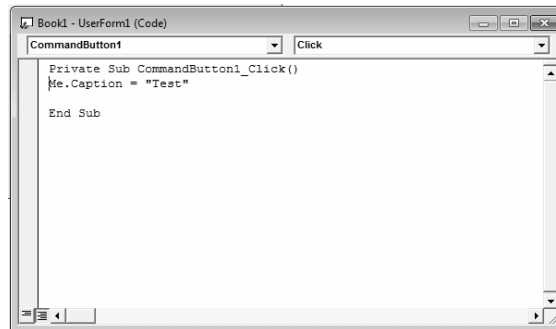
### 1.3.3 Jendela Kode

Jendela kode menampilkan kode yang berkaitan dengan item di jendela Project. Untuk melihat jendela kode ini, Anda bisa mengklik **View > Codes** atau klik F7 untuk menampilkan jendela kode ini.



*Gambar 1.8 Menu View > Code untuk menampilkan kode*

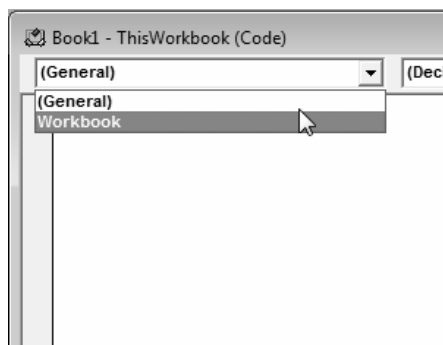
Umumnya, sebuah modul kode (standard, class, atau form) bisa berisi satu atau lebih prosedur. IDE dari jendela kode ini, bisa menampilkan satu prosedur saja atau semua prosedur secara bersamaan.



*Gambar 1.9 Jendela Kode*

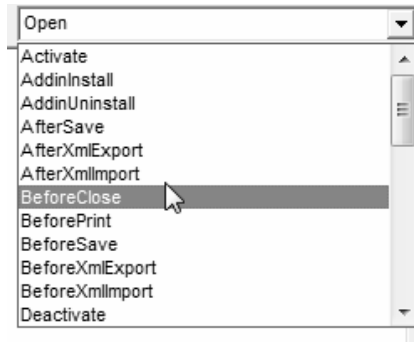
Di bagian atas jendela **Kode**, ada dua list box. Box Object berisi daftar objek, seperti form dan control yang berkaitan dengan proyek yang ada sekarang. Sementara yang satunya adalah kotak prosedur yang berisi daftar semua prosedur yang berkaitan dengan objek yang terpilih di kotak Object.

Konten selengkapnya dari box ini tergantung pada tipe objek yang terpilih di project explorer. Kalau workbook atau sheet yang terpilih, maka kotak Object hanya berisi dua entri, general, dan objek yang sedang dibuka.



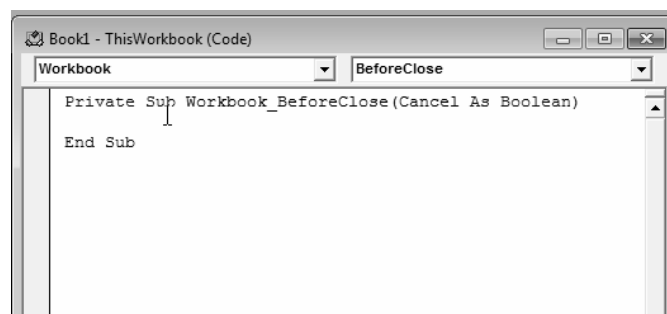
*Gambar 1.10 Ketika dipilih objek workbook maka yang terlihat adalah objek General dan Worksheet*

Kalau misalnya dipilih prosedur BeforeClose di kotak prosedur, Excel otomatis membuat kode yang akan menangani ketika objek ini hendak dimatikan.



**Gambar 1.11** Pengisian prosedur BeforeClose

Kalau sudah dipilih, otomatis di jendela kode akan menampilkan kode seperti berikut.

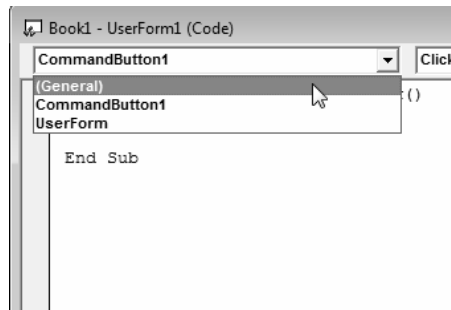


**Gambar 1.12** Menu BeforeClose

Kode ini menjelaskan kode yang akan menangani event ketika workbook akan ditutup.

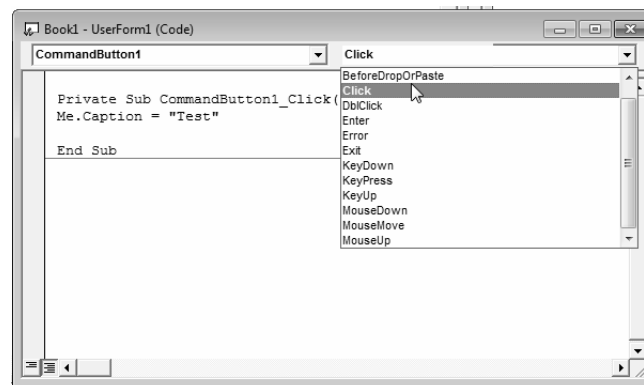
```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
'kode di sini untuk menangani event
End Sub
```

Ketika objek Form dipilih di project explorer, maka kotak object akan berisi daftar semua objek yang ada di form. Misalnya command button dan sebagainya.



**Gambar 1.13** Contoh object yang ditampilkan di list box

Prosedur yang dimilikinya pun akan berkaitan dengan objek yang dipilih, misalnya kalau untuk button, maka terlihat prosedur seperti click, DblClick dan sebagainya.



**Gambar 1.14** Prosedur untuk objek commandButton

### 1.3.4 Shortcut Navigasi

Ketika bekerja dengan IDE dari VBA, ada baiknya Anda mengenal beberapa shortcut yang dipakai di sini. Shortcut-shortcut navigasi ini akan memudahkan pekerjaan Anda dalam memprogram.

Berikut ini beberapa shortcut untuk navigasi:

1. F7

Membuka jendela kode.

2. F4  
Membuka jendela properties.
3. CTRL+R  
Membuka jendela proyek.
4. CTRL+G  
Membuka jendela intermediate.
5. ALT+F11  
Berpindah antara Excel dan IDE VB.

Sementara kalau ada di tampilan kode, shortcut berikut ini penting:

1. F1  
Menampilkan item help di bawah kursor.
2. SHIFT+F2  
Melihat definisi item di bawah kursor ini, jika kursor di atas fungsi tertentu atau subrutin, maka klik pada SHIFT+F2 akan menampilkan definisi dari prosedur tersebut.
3. CTRL+SHIFT+F2  
Kembali ke posisi terakhir dari editing yang sedang dilakukan.

Sementara ketika men-tracking melalui kode, Anda bisa menggunakan beberapa shortcut berikut ini:

1. F8  
Step into.
2. SHIFT+F8  
Step over.
3. CTRL+SHIFT+F8  
Step out.
4. CTRL+F8  
Menjalankan ke kursor.

5. F5  
Menjalankan.
6. CTRL+BREAK  
Break.
7. SHIFT+F9  
Quick watch.
8. F9  
Toggle breakpoint.
9. CTRL+SHIFT+F9  
Menghapus semua breakpoint.

## 1.4 Macro

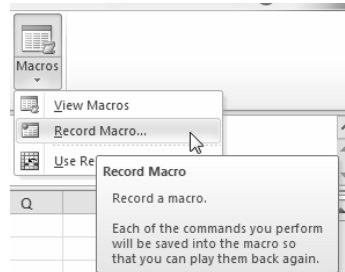
Macro adalah inti dari buku ini. Di awal sejarahnya, macro tersusun dari rangkaian keystroke dari keyboard yang direkam dan dialokasikan shortcut tertentu. Ketika user menjalankan shortcut yang bersangkutan, rekaman yang sudah disimpan akan dieksekusi. Sekarang ini, Macro sudah lebih canggih, Anda bisa membuat program Macro yang dapat melakukan tugas dari yang simpel hingga yang agak kompleks.

### 1.4.1 Merekam Macro

Excel memiliki kemampuan untuk merekam macro yang sangat sederhana. Ketika Anda mengklik pada tombol Record new macro dari ribbon Excel, maka Excel otomatis akan merekam langkah-langkah kita dan menyimpannya ke dalam kode program yang digenerate secara otomatis tanpa memiliki parameter.

Misalnya, apabila kita merekam macro yang melakukan Find & Replace, ikuti langkah-langkah berikut:

1. Siapkan workbook untuk menerima apa yang Anda lakukan.
2. Kemudian klik ribbon **View > Macros > Record Macro**.



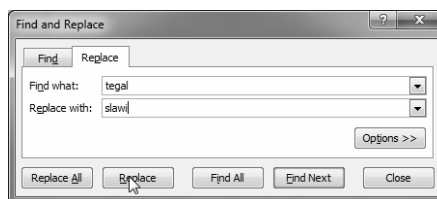
**Gambar 1.15 Menu untuk Record Macro**

3. Isikan nama macro di **Macro name**.
4. Isikan kode pintasan di **Shortcut key**.
5. Tentukan tempat menyimpan makro, kalau di file yang bersangkutan, pilih **This workbook**.
6. Kalau perlu, tentukan penjelasan mengenai macro ini di **Description**.



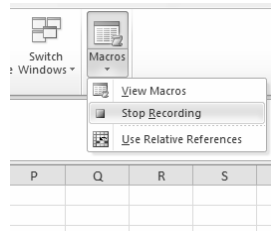
**Gambar 1.16 Mencari penjelasan mengenai macro di Description**

7. Klik OK, lalu lakukan action yang ingin direkam, misalnya **Find & Replace**, lalu ganti teks “tegal” menjadi teks “slawi”.



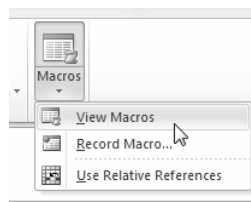
**Gambar 1.17 Action find and replace yang akan direkam**

8. Kalau sudah, klik pada **View > Macros > Stop recording**.



*Gambar 1.18 Menu untuk stop recording*

9. Sekarang Anda bisa melihat macro yang sudah terekam akan muncul di **Macros > View Macros**.



*Gambar 1.19 Macros > View Macros*

10. Terlihat nama makro yang sudah direkam ada di list yang ada.



*Gambar 1.20 Pengisian nama makro yang sudah direkam*

11. Untuk menjalankan makro, misalnya kondisi awal, ada beberapa sel yang punya teks = "tegal".



	A	B	C	D	E
1					
2					
3					
4					
5				tegal	
6				tegal	
7				adiwarna	
8					

**Gambar 1.21 Beberapa teks di sel dengan isi = “tegal”**

12. Kemudian jalankan dengan klik Run pada jendela Macro atau dengan mengklik shortcut yang sudah dibuat sebelumnya.



**Gambar 1.22 Klik tombol Run untuk menjalankan macro**

13. Maka otomatis teks di seluruh workbook yang bertuliskan “tegal” akan berubah menjadi “slawi”.

	A	B	C	D	E
1					
2					
3					
4					
5				slawi	
6				slawi	
7				adiwarna	
8					
9					

**Gambar 1.23 Teks yang sudah dibuat akan bertuliskan “slawi”**

14. Untuk melihat kode dari macro yang sudah dibuat, pilih macro tersebut di jendela Macro, kemudian klik Edit.



**Gambar 1.24** Klik pada tombol Edit sambil memilih makro untuk melihat kode yang dihasilkan macro

15. Hasilnya, muncul tampilan kode dari program.



**Gambar 1.25** Tampilan kode dari program

16. Anda bisa melihat nama makro menjadi nama prosedur.

```
Sub caridanGanti()
'
' caridanGanti Macro
'
' Keyboard Shortcut: Ctrl+g
'
Cells.Replace What:="tegal", Replacement:="slawi",
LookAt:=xlPart, _
SearchOrder:=xlByRows, MatchCase:=False,
SearchFormat:=False, _
ReplaceFormat:=False
End Sub
```

Dalam beberapa situasi, fasilitas macro recorder ini sangat penting karena bisa digunakan sebagai alat bantu untuk belajar. Jika Anda tidak tahu bagaimana sintaks untuk melakukan sesuatu, maka Anda bisa mencoba merekamnya menggunakan macro recorder, dan kemudian melihat kodenya untuk disalin di program Anda. Contohnya ketika Anda hendak membuat pivot table.

Tapi, recorder macro ini hanya bisa dipakai untuk merekam situasi yang sederhana, dan bukan yang terlalu kompleks. Sementara program Excel pada umumnya menuntut situasi yang kompleks, yang terlalu rumit kalau mau direkam dengan fasilitas ini.

## **1.5 Mengetahui Sintaks Bahasa Pemrograman VBA**

Untuk bisa mempraktikkan pembuatan proyek macro di VB, Anda perlu mengetahui tata cara penulisan sintaks bahasa pemrograman VBA. Ibaratnya kalau mau memerintahkan komputer untuk melakukan tugas tertentu, Anda harus tahu cara memerintahkannya. Di subbab ini akan dibahas dasar-dasar dari pemrograman VBA yang menjadi dasar untuk semua pemrograman MS Office.

### **1.5.1 Komentar**

Komentar adalah bagian yang sangat penting di bahasa pemrograman manapun. Karena fungsinya sangat krusial, yaitu untuk memberitahukan programmer apa sebenarnya maksud kode-kode yang dituliskan.

Ini karena bahasa pemrograman walaupun tingkat tinggi sekalipun yang seharusnya mudah dipahami, tetaplah “kaku” dan tidak seperti bahasa yang kita pakai sehari-hari, sehingga komentar ini membantu memudahkan programmer untuk membaca program yang dibuatnya.

Mungkin programmer kalau baru saja menuliskan kode program akan mudah mengingat maksud program yang dibuatnya, tapi kalau kode program sudah berumur enam bulan ke atas, biasanya sudah banyak lupa apa yang dituliskan kemarin.

Cara untuk memberikan komentar di VBA mirip dengan di Visual Basic, yaitu dengan tanda petik atas satu. Contoh kodenya seperti berikut:

```
Private Sub CommandButton1_Click()  
'ini tidak diperhitungkan  
'ini komentar  
Dim a As Integer  
a = 1  
End Sub
```

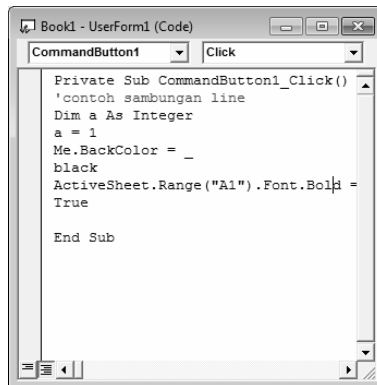
Kode di atas tidak diperhitungkan karena ada tanda ' di sebelah kiri dari tiap baris. Komentar juga penting untuk tracking kesalahan, karena Anda bisa memberi catatan tentang apa yang terjadi selama debugging.

### 1.5.2 Penyambungan Baris

Sebuah program bisa terdiri atas baris yang amat panjang, di mana ini bisa membuat susah untuk dipahami. Untuk itu Anda bisa memecah baris yang panjang ke baris selanjutnya, tapi dengan menggunakan tanda underscore \_ sehingga baris tersebut masih dianggap sebagai satu baris.

```
Private Sub CommandButton1_Click()  
'contoh sambungan line  
Dim a As Integer  
a = 1  
Me.BackColor = _  
black  
ActiveSheet.Range("A1").Font.Bold = _  
True  
End Sub
```

Tiap baris yang ada tanda \_ berarti masih dianggap satu baris dengan baris di bawahnya.



**Gambar 1.26** Contoh penulisan kode pemisah baris dengan tanda \_

### 1.5.3 Konstanta

Bahasa VBA memiliki dua jenis konstanta, yaitu konstanta literal dan konstanta simbolis. Konstanta literal atau konstanta saja merupakan nilai tertentu, seperti angka, tanggal, atau teks yang tidak berubah nilainya dan digunakan sesuai nilai yang ditulisnya.

```

Private Sub CommandButton1_Click()
'contoh konstanta literal
Dim kon1 As String
kon1 = "Semarang"
End Sub

```

Sementara konstanta simbolis atau sering disebut constant adalah nama dari konstanta literal. Misalnya:

```

Private Sub CommandButton1_Click()
'contoh konstanta simbolis
Const LokData = "D:\data"
End Sub

```

Enaknya menggunakan konstanta adalah kalau nilai konstanta tersebut mau diubah, tinggal diubah di konstanta saja, tidak perlu meng-update keseluruhan kode program. Jadi, kalau misalnya Lokasi data mau diubah ke "D:\databaru", maka yang diubah tinggal const LokData saja, tanpa harus mengupdate seluruh kode program.

Biasanya konstanta didefinisikan di bagian awal kode. Ini membuat lebih terbaca dan perawatan menjadi lebih mudah.

## 1.6 Variabel dan Tipe Data

Sebuah variabel adalah komponen yang sangat penting dalam pemrograman. Karena nilai-nilai data yang akan digunakan di program disimpan dalam bentuk variabel ini.

Variabel adalah lokasi di memori yang akan menyimpan nilai dengan tipe tertentu. Berbeda dengan konstanta, nilai dari variabel ini bisa berubah sepanjang siklus hidup program. Karena itulah namanya variabel (vary = berubah).

Di VBA, tiap variabel memiliki tipe data khusus, yang mengindikasikan tipe data yang disimpannya. Contohnya variabel yang menyimpan string teks memiliki tipe data String dan disebut variabel string.

Sementara variabel yang menyimpan data integer (bilangan genap) memiliki tipe data integer dan disebut integer.

Tabel berikut menampilkan beberapa tipe data variabel beserta jumlah memori yang dipakainya.

Tipe	Ukuran di Memori	Cakupan Nilai
Byte	1 byte	0 - 255
Boolean	2 byte	True atau False
Integer	2 byte	-32,768 - 32,767
Long (long integer)	4 byte	-2,147,483,648 - 2,147,483,647
Single (single-precision real)	4 byte	-3.4E38 - 3.4E38
Double (double-precision real)	8 byte	-1.8E308 - 4.9E324
Currency (scaled integer)	8 byte	-922,337,203,685,477.5808 - 922,337,203,685,477.5807
Date	8 byte	1/1/100 - 12/31/9999
Object	4 byte	Semua Object reference

String	Variable: 10 bytes + panjang string; Fixed: panjang string	Sekitar 2 miliar (65,400 untuk Win 3.1) Fixed length: hingga 65,400
Variant	16 bytes untuk angka	Number: sama Doublestring: sama seperti string
User-defined	Bervariasi	

### 1.6.1 Deklarasi Variabel

Mendeklarasikan variabel artinya mendefinisikan tipe datanya. Variabel di VBA dideklarasikan sama seperti di VB, yaitu menggunakan kata kunci DIM. Atau bisa juga dengan kata kunci Private dan Public.

Berikut ini contoh beberapa deklarasi untuk variabel:

```
Dim Nama As String
Dim Tanggal As Date
Dim Angka As Integer
Dim TinggiBadan As Single
Dim Uang As Currency
Dim wbk As Workbook
Dim ch As Chart
```

Sintaks yang dipakai untuk mendeklarasikan ini adalah:

```
Dim namaVariabel As DataType
```

Jika ada variabel khusus digunakan tanpa mendeklarasikan terlebih dahulu, atau dideklarasikan tanpa mendeklarasikan tipe variabelnya, maka VBA akan menganggapnya sebagai variabel dengan tipe data variant. Contohnya:

```
Dim umur
```

Tapi ini akan memboroskan memori Anda, karena variant membutuhkan memori lebih banyak dibandingkan tipe variabel lainnya. Contohnya variabel integer membutuhkan 2 byte, sementara variant yang menyimpan nilai yang sama akan membutuhkan 16 byte.

Anda bisa mendefinisikan beberapa variabel dengan satu kali Dim seperti berikut:

`Dim Umur As Integer, Nama As String`

Sementara kalau deklarasi seperti berikut salah:

`Dim Umur, Nama As String`

Karena kalau dideklarasikan yang kedua, hanya Nama yang berjenis string, sementara Umur akan berjenis variant.

Anda juga bisa memberitahukan VBA tipe variabel dengan menambahkan karakter khusus ke nama variabel. Berikut ini akhiran dari variabel yang menentukan jenis variabel.

Akhiran	Tipe
%	Integer
&	Long
!	Single
#	Double
@	Currency
\$	String

Misalnya kode menjadi:

`Dim Nama$`

`Nama$ = "Sherly"`

Walaupun Excel memungkinkan deklarasi variabel dan konstanta diletakkan di bagian manapun di prosedur, tapi disarankan melakukan pendeklarasian di bagian awal prosedur. Ini membuat program lebih mudah dibaca dan di-tracking.

### 1.6.2 Tipe Data Numerik

Jadi di VBA, yang termasuk tipe data numerik adalah integer, long, single, double dan currency. Long ini sering juga disebut long integer.



### 1.6.3 Tipe Data Boolean

Variabel boolean adalah variabel yang isinya hanya satu dari dua ini, yaitu True atau False. Walaupun sangat sederhana, tapi peranannya sangat penting. Yaitu untuk uji logika dan mengarahkan ke mana flow program.

### 1.6.4 Tipe Data String

String adalah urutan karakter. Sebuah string bisa berisi karakter biasa (huruf, digit, dan tanda baca), maupun karakter kontrol khusus seperti vbCrLf (karakter line feed) atau vbTab (karakter tab). Konstanta string ini biasanya diapit dengan tanda petik dua.

Ada dua jenis tipe string di vba, yaitu yang panjangnya tetap, dan yang panjangnya variabel. Untuk mendeklarasikan string yang panjangnya tetap, caranya seperti berikut:

```
Dim namaVariabel As String * panjang_string
```

Misalnya:

```
Dim nama1 As String * 10
```

Sementara string yang panjangnya bervariasi bisa menyimpan string dengan panjang yang berubah-ubah. Cara mendeklarasikannya seperti berikut:

```
Dim namaVariabel as String
```

Contohnya:

```
Dim nama2 As String
```

### 1.6.5 Tipe Data Date/Tanggal

Variabel date membutuhkan tempat simpan sebesar 8 byte dan disimpan dalam bentuk angka pecahan desimal (floating point) yang berkisar dari 1 Januari 100 hingga 31 Desember 9999.

Cara mendefinisikan tipe data date adalah:

```
Dim tgl As Date  
tgl = #1/2/2012#  
tgl = "January 12, 2001"
```

### 1.6.6 Tipe Data Variant

Tipe data variant ini akan menyimpan semua jenis data, kecuali data string dengan panjang yang tetap. Tetapi tipe data variant diusahakan jangan sering dipakai karena memakan memori dengan ukuran yang sangat besar.

### 1.6.7 Tipe Data Objek Excel

VBA Excel memiliki banyak tipe data tambahan yang termasuk kategori tipe data objek Excel. Berikut ini beberapa di antaranya.

Objek yang berkaitan dengan chart:

- Axis
- ChartTitle
- Legend
- Series
- AxisTitle
- DataLabel
- LegendEntry
- SeriesCollection
- Chart
- DataTable
- LegendKey
- TickLabels
- ChartArea
- Floor
- PlotArea
- Walls
- ChartColorFormat
- Gridlines
- Point

Objek yang berkaitan dengan Pivot table:

- PivotCache
- PivotField
- PivotFormula
- PivotItem
- PivotTable

Objek umum:

- Comment
- Font
- Range
- Workbook
- FillFormat
- Outline
- Sheets
- Worksheet
- Filter
- PageSetup
- Window
- WorksheetFunction

Contoh cara pendeklarasian tipe objek Excel ini seperti berikut:

- Dim FileExcel As Workbook
- Dim Sheet1 As Worksheet
- Dim Gambar1 As Chart
- Dim Ax1 As axis
- Dim fieldPivot1 As PivotField

### 1.6.8 Array

Variabel array/larik merupakan kumpulan variabel yang menggunakan nama yang sama, namun dibedakan menurut nilai indeksinya. Misalnya untuk menyimpan nilai baris pertama dari worksheet, kita bisa mendeklarasikan variabel seperti berikut:

```
Dim Cell(1 To 100) As Range
```

Tidak ada objek sel di Excel, sebuah sel adalah objek range khusus.

Variabel array Cell di atas memiliki ukuran 100. Batas bawahnya adalah 1 dan batas atasnya adalah 100. Tiap variabel adalah variabel range, yaitu variabel dengan tipe objek range:

```
Cell(1), Cell(2), ..., Cell(100)
```

Apabila indeks pertama dihilangkan, misalnya dengan deklarasi seperti berikut:

```
Dim Cell(100) As Range
```

VBA otomatis akan mengeset indeks ke-0 sehingga ukuran array akan menjadi 101. Anda bisa memanggil isi-isi range, dengan menggunakan loop For i=1 to 100. Contoh penggunaan array dalam program seperti berikut:

```
For i = 1 To 100
    Set Cell(i) = Cells(i,i)
    Cell(i).Font.Bold = True
Next i
```

## 1.7 VBA Operator

VBA menggunakan beberapa operator dan relasi untuk mengoperasikan data-data, baik variabel atau konstanta. Anda perlu memahami beberapa operator dan relasi VBA sebelum bisa melakukan pemrograman.

Tipe	Nama	Simbol
Operator aritmetika	Penambahan	+
	Pengurangan	-
	Pembagian	/

	Pembagian dengan hasil integer/genap	\
	Eksponensial	^
	Modulus	Mod
Operator string	Concatenation/Penyambungan	&
Operator logik	AND	AND
	OR	OR
	NOT	NOT
Relasi Perbandingan	Sama dengan	=
	Kurang dari	<
	Lebih dari	>
	Kurang dari atau sama dengan	<=
	Lebih dari atau sama dengan	>=
	Tidak sama dengan	<>

Operator Mod akan mengembalikan sisa hasil pembagian.  
Contohnya:

8 Mod 3

Fungsi di atas akan mengembalikan nilai 2, karena sisa pembagian 8/3 adalah 2.

Contoh untuk menyambung/concatenate:

"Tegal " & "Jakarta"

Akan sama dengan:

"Tegal Jakarta"

## 1.8 Fungsi dan Subrutin

VBA memungkinkan dua jenis prosedur, yaitu fungsi dan subrutin (prosedur biasa). Apa bedanya? Perbedaannya fungsi mengembalikan nilai, sementara subrutin tidak mengembalikan nilai.

Bentuk deklarasi fungsi memiliki bentuk seperti berikut:

```
[Public atau Private] Function NamaFungsi (Param1 As TipeData1, _  
    Param2 As TipeData2,...) As TipeKembalian
```

Perhatikan bahwa kita tidak hanya mendeklarasikan tipe data dari tiap parameter di fungsi, tapi juga tipe kembaliannya. Kalau tidak didefinisikan, maka VBA akan mendeklarasikan item sebagai varian.

Adapun Public atau Private berguna untuk mengindikasikan skop dari fungsi, sebagaimana dideklarasikan variabel.

Contoh fungsi seperti berikut:

```
Public Function TambahkanSatu (Value As Integer) As Integer  
    TambahkanSatu = Value + 1  
End Function
```

Untuk memanggil fungsi, kita tinggal meletakkan panggilan dengan mengetikkan nama fungsi di lokasi yang diinginkan. Contohnya adalah kalau mau memasukkan di messagebox adalah:

```
MsgBox "Menambahkan 1 ke angka 5 menghasilkan: " &  
TambahanSatu(5)
```

Ini akan menghasilkan nilai 6, karena fungsi TambahkanSatu akan menambahkan value di dalam kurung dengan angka satu. Untuk mengembalikan nilai dari fungsi, kita harus memberikan nama fungsi ke value yang ingin dikeluarkan di dalam body dari fungsi.

Berikut ini contoh lainnya:

```
Function HitungSel() As Variant  
    ' Mengembalikan jumlah sel di seleksi yang ada  
  
    If TypeName(Selection) = "Range" Then  
        HitungSel = Selection.Count  
    Else  
        HitungSel = "Tidak bisa"  
    End If  
End Function
```

Fungsi ini akan mengembalikan jumlah sel di selection yang dipilih. Jika selection memilih objek yang lain seperti chart, maka fungsi akan mengembalikan pesan "Tidak bisa".

Karena nilai kembalian bisa berupa angka atau string maka kita mendeklarasikan hasil kembalian sebagai variant.

Karena fungsi ini mengembalikan value, maka Anda tidak bisa memanggilnya secara langsung dari kotak dialog macro yang

muncul ketika merekam sebelumnya, tapi Anda perlu membungkusnya dalam bentuk subrutin atau prosedur.

Adapun subrutin memiliki bentuk sintaks seperti berikut:

```
[Public atau Private] Sub NamaSubRutin(Param1 As TipeData1, _  
    Param2 As TipeData2,...)
```

Sintaks ini mirip dengan deklarasi fungsi, hanya saja tidak memiliki `As TipeKembalian`. Selain itu ada juga yang tidak bisa digunakan di dalam ekspresi. Untuk memanggil subrutin dengan nama `SubrutinA` misalnya, maka kita bisa menuliskan:

```
Call SubrutinA(parameter, . . .)
```

Atau lebih simpel:

```
SubrutinA parameter, . . .
```

Pastikan bahwa semua parameter harus diberi tanda kurung ketika menggunakan `Call`, tapi kalau satunya tidak perlu.

### 1.8.1 Parameter dan Argumen

Parameter dan argumen fungsinya penting dalam fungsi atau subrutin. Contohnya pada subrutin seperti berikut ini:

```
Sub TampilkanNama(sNama As String)  
    MsgBox "Nama Anda adalah" & sNama  
End Sub
```

Untuk memanggil subrutin ini, kita tinggal mengetikkan seperti berikut:

```
TampilkanNama "Sherly"
```

Atau:

```
Call TampilkanNama ("Sherly")
```

Variabel `sNama` dideklarasikan prosedur:

```
Sub TampilkanNama(sNama As String)
```

Disebut parameter dari prosedur. Panggilan prosedur harus berisi variabel string atau string literal yang direpresentasikan oleh variabel `sNama` di prosedur ini. Parameter yang dimasukkan di prosedur ini disebut argumen. Sehingga pada contoh di atas, argumennya adalah string "Sherly".

### 1.8.2 Argumen Opsional

Di VBA, argumen-argumen pada sebuah prosedur bisa ditentukan sebagai opsional menggunakan kata kunci Optional. Contoh penggunaan argumen opsional seperti berikut:

```
Sub UbahFormat(NamaFont As String, _  
               Optional UkuranFont As Variant)  
  
    ' Ubah nama font  
    Selection.Font.Name = NamaFont  
  
    ' Ubah ukuran font jika argumen dimasukkan  
    If Not IsMissing(UkuranFont) Then  
        Selection.Font.Size = CInt(UkuranFont)  
    End If  
End Sub
```

### 1.8.3 Keluar dari Prosedur

VBA menyediakan statement Exit Sub dan Exit Function statements. Ini dipakai untuk keluar dari prosedur atau fungsi sebelum prosedur atau fungsi tersebut berhenti secara alami.

Misalnya kalau nilai dari parameter tidak cocok, kita ingin menampilkan warning ke user dan program keluar. Contoh penggunaan Exit statement ini adalah:

```
Sub TampilkanNama (sNama As String)  
    If sNama = "" then  
        MsgBox "Tolong masukkan nama."  
        Exit Sub  
    End If  
    MsgBox "Nama yang dimasukkan adalah " & sNama  
End Sub
```

### 1.8.4 Prosedur Public dan Private

Sebagaimana variabel dan konstanta memiliki skope, begitu pula prosedur juga memiliki skope. Kita bisa mendeklarasikan prosedur menggunakan kata kunci Public atau Private. Contohnya:

```
Public Function TambahkanSatu(i As Integer) As Integer
```

Atau:

```
Private Function TambahkanSatu(i As Integer) As Integer
```



Apa perbedaan skope dari Public atau Private?

Prosedur Private hanya bisa dipanggil dari dalam modul tempatnya didefinisikan, sementara prosedur Public bisa dipanggil dari modul manapun di proyek.

Perhatikan bahwa jika kata kunci Public atau Private dihilangkan dari deklarasi prosedur, maka prosedur ini akan dianggap Public.

## 1.9 Fungsi dan Statement Penting yang Sudah Built In

VBA memiliki banyak fungsi dan statement yang built ini. Anda perlu sekali memahaminya karena ini akan membantu Anda dalam mempermudah pemrograman. Dari semua fungsi dan statement yang sudah built in, ada beberapa yang sering dipakai.

Berikut ini beberapa fungsi bawaan/built in dari VBA Excel:

- Abs
- DoEvents
- IsEmpty
- Right
- Array
- Environ
- IsError
- RightB
- Asc
- EOF
- IsMissing Rnd
- AscB
- Error
- IsNull Round\*
- AscW
- Exp
- IsNumeric
- RTrim
- Atn

- FileAttr
- IsObject
- Second
- CallByName\*
- FileDateTime
- Join\*
- Seek
- Cbool
- FileLen
- Lbound
- Sgn
- Cbyte
- Filter\*
- Lcase
- Shell
- Ccur
- Fix
- Left
- Sin
- Cdate
- Format
- LeftB
- SLN
- CDbl
- FormatCurrency\*
- Len
- Space
- Cdec
- FormatDateTime\*
- LenB
- Spc
- Choose

- FormatNumber\*
- LoadPicture
- Split\*
- Chr
- FormatPercent\*
- Loc
- Sqr
- ChrB
- FreeFile
- LOF
- Str
- ChrW
- FV
- Log
- StrComp
- Cint
- GetAllSettings
- Ltrim
- StrConv
- CLng
- GetAttr
- Mid
- String
- Command
- GetAutoServerSettings
- MidB
- StrReverse\*
- Cos
- GetObject
- Minute
- Switch
- CreateObject

- GetSetting
- MIRR
- SYD
- CSng
- Hex
- Month
- Tab
- CStr
- Hour
- MonthName\*
- Tan
- CurDir
- Iif
- MsgBox
- Time
- Cvar
- IMESStatus
- Now
- Timer
- CVDDate
- Input
- Nper
- TimeSerial
- CVErr
- InputB
- NPV
- TimeValue
- Date
- InputBox
- Oct
- Trim
- DateAdd

- InStr
- Partition
- TypeName
- DateDiff
- InStrB
- Pmt
- Ubound
- DatePart
- InstrRev\*
- PPmt
- UCase
- DateSerial
- Int
- PV
- Val
- DateValue
- Ipmt
- QBColor
- VarType
- Day
- IRR
- Rate
- WeekdayWeekdayName\*
- DDB
- IsArray
- Replace\*
- Year
- Dir
- IsDate
- RGB

Fungsi yang diberi tanda bintang (\*) adalah fungsi dari VBA versi terakhir. Dari beberapa fungsi dan prosedur di atas, tidak semua

fungsi penting dan lazim digunakan. Kita akan membahas fungsi-fungsi yang paling sering dipakai.

### 1.9.1 Fungsi MsgBox

MsgBox adalah fungsi untuk menampilkan message box alias kotak pesan. Fungsi ini akan menampilkan kotak pesan dan menunggu input dari user yang akan mengklik tombol di sini.

Sintaks yang akan digunakan adalah:

```
MsgBox(prompt [, button] [, title])
```

Ini bukanlah sintaks lengkap dari fungsi ini, karena ada beberapa parameter yang dihilangkan. Untuk melihat lebih detail, Anda bisa melihat dokumentasi help dari VBA.

Prompt adalah parameter string yang berisi pesan yang akan ditampilkan di dialog. Perhatikan bahwa tampilan pesan multiline bisa dibuat menggunakan konstanta vbCrLf di dalam pesan.

Button adalah parameter bertipe long yang menentukan properti dari berbagai nilai message box. Properti ini akan menentukan angka dan tipe button yang akan ditampilkan, style ikon yang akan dipakai, identitas button default, serta tipe dari messagebox.

Contohnya kotak dialog sistem akan tetap di atas dari semua window dan akan meng-capture fokus input dari sistem secara keseluruhan. Sementara kotak dialog box yang model aplikasi akan di atas dari semua jendela aplikasi saja, dan hanya akan meng-capture fokus dari aplikasi saja.

Beberapa tipe button yang didefinisikan di VbMsgBoxStyle adalah:

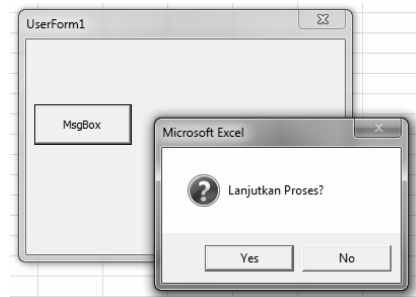
Tujuan	Konstanta	Nilai	Deskripsi
Tipe Button	vbOKOnly	0	Menampilkan tombol OK saja.
	vbOKCancel	1	Menampilkan tombol OK dan Cancel.
	vbAbortRetryIgnore	2	Menampilkan tombol Abort, Retry, dan Ignore.

	vbYesNoCancel	3	Menampilkan tombol Yes, No, dan Cancel buttons.
	vbYesNo	4	Menampilkan tombol Yes dan No.
	vbRetryCancel	5	Menampilkan tombol Retry dan Cancel.
Tipe Icon	vbCritical	16	Menampilkan ikon pesan kritis.
	vbQuestion	32	Menampilkan ikon query warning.
	vbExclamation	48	Menampilkan ikon pesan warning.
	vbInformation	64	Menampilkan ikon pesan Information.
Tombol Default	vbDefaultButton1	0	Tombol pertama adalah default.
	vbDefaultButton2	256	Tombol kedua adalah default.
	vbDefaultButton3	512	Tombol ketiga adalah default.
	vbDefaultButton4	768	Tombol keempat adalah default.
Modality	vbApplicationModal	0	Application modal message box.
	vbSystemModal	4096	System modal message box.

Contohnya Anda bisa membuat form, kemudian di event click dari command button-nya, Anda bisa memberi kode seperti berikut:

```
Private Sub CommandButton1_Click()
MsgBox "Lanjutkan Proses?", vbQuestion + vbYesNo
End Sub
```

Apabila dijalankan, akan muncul message box seperti berikut ini.



**Gambar 1.27 Tampilan message box yang dibuat menggunakan fungsi MsgBox**

Ada parameter title yang menunjukkan parameter string yang ditampilkan di title bar dari message box. Jika ini dihilangkan, maka yang akan ditampilkan adalah tulisan “Microsoft Excel”.

Fungsi MsgBox dapat mengembalikan fungsi yang menjelaskan button mana yang dipilih. Nilai angka dari pengembalian fungsi ini seperti berikut.

Konstanta	Nilai	Deskripsi
vbOK	1	Tombol OK ditekan
vbCancel	2	Tombol Cancel ditekan
vbAbort	3	Tombol Abort ditekan
vbRetry	4	Tombol Retry ditekan
vbIgnore	5	Tombol Ignore ditekan
vbYes	6	Tombol Yes ditekan
vbNo	7	Tombol No ditekan

## 1.9.2 Fungsi InputBox

Kalau fungsi MsgBox digunakan untuk menampilkan pesan kepada user, maka fungsi InputBox ini digunakan untuk menerima masukan dari user. Sintaks dari InputBox yang paling sering dipakai adalah:

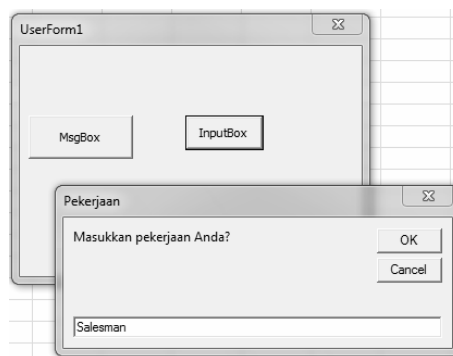


```
InputBox(prompt [, title] [, default])
```

Di mana prompt di atas adalah pesan yang akan ditampilkan di InputBox. Title adalah judul untuk input box, dan default adalah teks awal yang ditampilkan di input box.

```
sNama= InputBox("Masukkan pekerjaan Anda?", "Pekerjaan",  
"Salesman")
```

Kode di atas akan menampilkan tampilan seperti berikut.



**Gambar 1.28 Contoh inputBox yang dimasukkan dari button InputBox**

Fungsi InputBox ini akan mengembalikan string yang dimasukkan oleh user ke textbox. Pada contoh di atas, variabel string sNama akan berisi string hasil pemasukan user. Kalau Anda mau mengambil inputan berupa angka, Anda bisa mengonversi string yang dikembalikan ke bentuk angka dengan fungsi Val.

### 1.9.3 Beep

Statement ini sangat simpel, cuma mengetikkan Beep. Ini akan menghasilkan suara dari komputer yang membuat user bisa fokus pada hal tertentu sesuai keinginan programmer.

```
Private Sub CommandButton3_Click()  
Beep  
End Sub
```

## 1.10 Statement Control

Untuk bisa mengendalikan kontrol pemrograman, kita harus mengenal beberapa sintaks untuk mengatur flow dari program. Ini akan mengarahkan ke mana program akan dieksekusi.

### 1.10.1 Statement If-Then

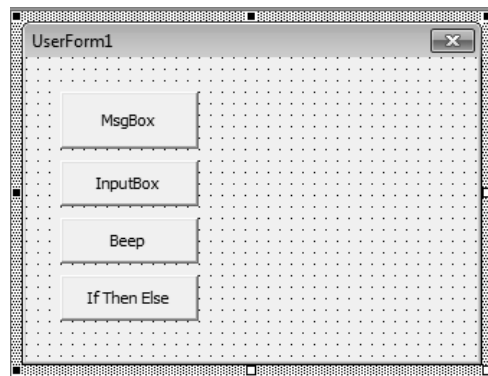
Statement If Then digunakan untuk pengaturan kondisional. Sintaks dari If Then adalah:

```
If kondisi Then  
    ' statement di sini. . .  
ElseIf kondisiLainnya Then  
    ' statement di sini. . .  
Else  
    ' statement lain di sini. . .  
End If
```

Kita bisa membuat statement lebih dari ini.

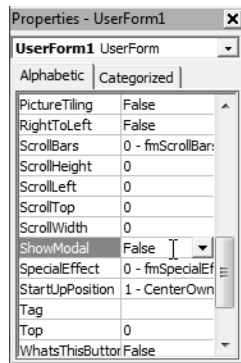
Oke langsung praktik, kita akan menerapkan statement If Then ini untuk membuat pola warna yang berselingan untuk baris genap dan ganjil seperti berikut:

1. Masukkan button dengan caption = “If Then else” di form.



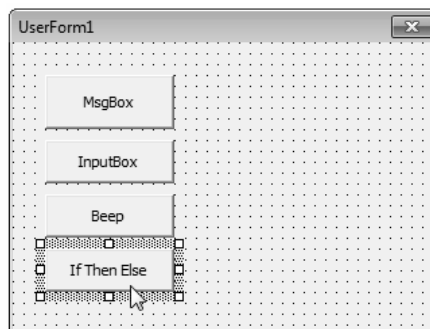
*Gambar 1.29 Memasukkan form If Then Else*

2. Set properti ShowModal dari form menjadi False agar Anda bisa menyeleksi sel ketika program ini dijalankan.



**Gambar 1.30** Set properti ShowModal menjadi False

3. Klik 2x pada button. Untuk memberikan kode ke dalam prosedur ButtonClick dari button ini.



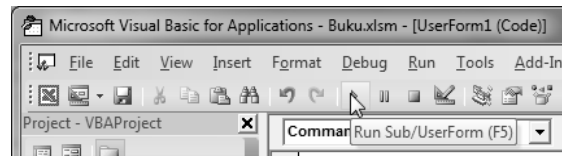
**Gambar 1.31** Klik 2x pada button If Then Else

4. Kode yang dimasukkan seperti berikut:

```
Private Sub CommandButton4_Click()
Dim pilihanSel As Range
'memilih sel genap atau ganjil,
'kemudian mengeset warna untuk tiap baris

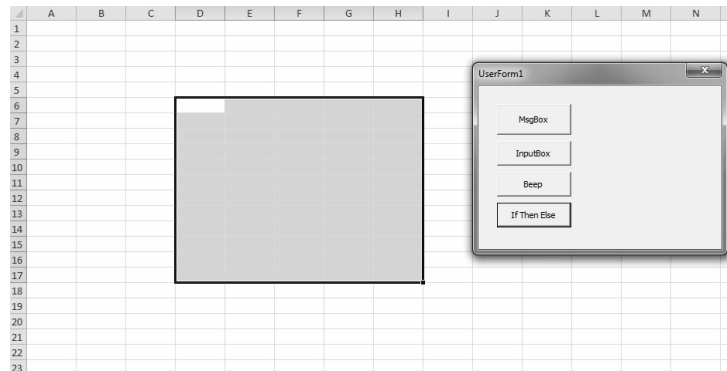
For Each pilihanSel In Selection.Cells
    If (pilihanSel.Row Mod 2) = 1 Then
        ' ganjil
        pilihanSel.Interior.ColorIndex = 2
    Else
        ' genap
        pilihanSel.Interior.ColorIndex = 1
    End If
Next
End Sub
```

5. Kemudian jalankan program dengan klik pada tombol **Run**.



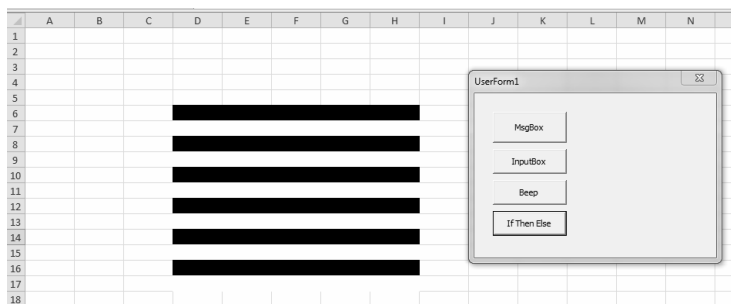
*Gambar 1.32 Jalankan program dengan klik Run*

6. Anda bisa memilih range tertentu.



*Gambar 1.33 Memilih range tertentu*

7. Kemudian klik pada tombol **If Then Else** untuk melihat pengaruh statement If Then Else terhadap range dari sel yang Anda pilih sebelumnya. Range genap menjadi gelap, sementara range ganjil menjadi tidak gelap.



*Gamba 1.34 Range ganjil dan genap menjadi berwarna beda*

### 1.10.2 For Loop

Statement For...Next adalah statement yang memudahkan Anda membuat pengulangan untuk melakukan loop pada beberapa blok kode. Loop pertama adalah For loop. Sintaks yang dipakai seperti berikut:

```
For konter = start To end
    ' Kode blok di sini. . .
Next konter
```

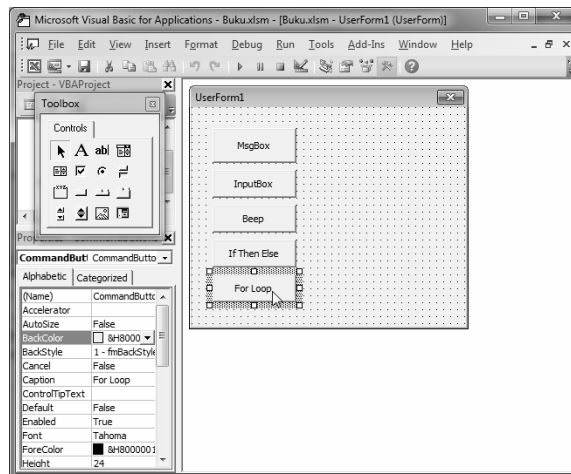
Pertama kali blok kode dieksekusi, variabel konter akan bernilai awal, sementara kalau sudah loop, maka konter akan ditambahkan dengan satu. Ketika konter sudah melebihi nilai akhir, maka blok kode tidak akan dieksekusi lagi.

Jadi, blok kode akan dieksekusi berulang sebanyak  $\text{konter\_akhir} - \text{konter\_awal} + 1$ .

Langsung praktik saja. Berikut ini contoh penggunaan For Loop untuk mencari apakah ada sel yang berisi tanggal/date di range dan kemudian memberi warna dan background yang berbeda untuk sel yang berisi date tersebut.

Caranya seperti berikut:

1. Tambahkan button dengan caption “For Loop” di Form.



**Gambar 1.35** Tambahkan button dengan caption For Loop

- Isikan kode berikut di dalamnya, kode ini akan memilih apakah ada sel yang berisi tanggal/date, dan kemudian memberikan warna font dan background ke sel yang berisi date tersebut.

```
Private Sub CommandButton5_Click()
Dim i As Integer
Dim pilihanSel As Range

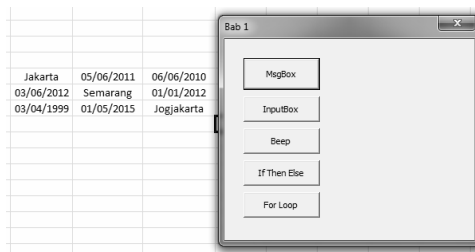
For i = 1 To Selection.Count
' Mendefinisikan seleksi
Set pilihanSel = Selection.Cells(i)

' Memberi warna dan background
' kalau data di seleksi berupa date

If IsDate(pilihanSel) Then
pilihanSel.Font.ColorIndex = 2
pilihanSel.Interior.ColorIndex = 1
End If
Next i

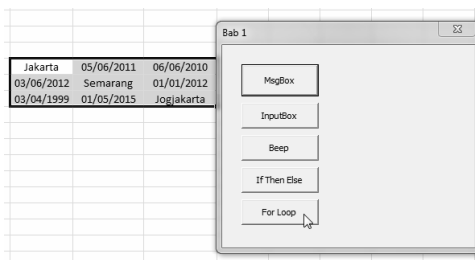
End Sub
```

- Kalau dijalankan program, user membuat sel-sel yang acak, ada teks dan ada date.



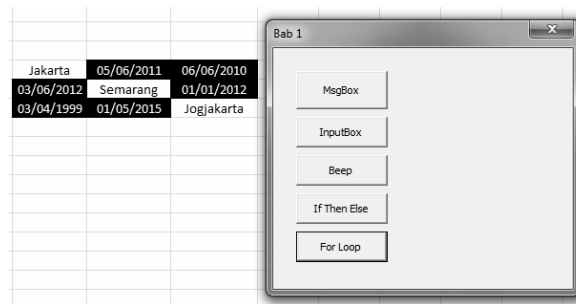
**Gambar 1.36** User membuat isi sel yang acak

- Seleksi pada range sel tersebut, dan klik tombol **For Loop**.



**Gambar 1.37** Klik tombol **For Loop** pada seleksi

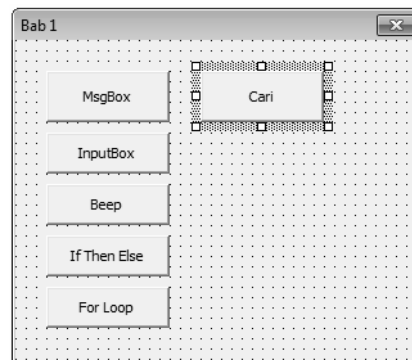
5. Maka sel-sel yang bertipe Date, akan berisi teks yang berbeda, dan memiliki background yang berbeda.



**Gambar 1.38** Sel yang berisi tanggal akan berwarna beda

Contoh praktik kedua penggunaan For Loop adalah untuk mencari teks di sel tertentu (misalnya kolom A) dari Excel. Ikuti langkah-langkah praktik berikut untuk membuat contoh penggunaan For Loop kedua:

1. Buat sebuah button dan ganti caption-nya dengan “Cari”.



**Gambar 1.39** Masukkan button “Cari” di form

2. Kemudian klik 2x pada button tersebut dan isikan kode berikut:

```
Private Sub CommandButton6_Click()
Dim sNama As String
sNama = InputBox("Teks Apa Yang Mau Dicari?", "Pencarian",
"isikan")
Me.Cari_String (sNama)
End Sub
```

3. Pada kode di atas, ada panggilan ke fungsi Cari\_string(). Oleh karena itu, di bawah prosedur pada form, isikan kode berikut:

```
Sub Cari_String(sTeksPencarian As String)

    Dim i As Integer
    ' Integer yang mau dipakai untuk loop 'For'
    Dim iHasilPencarian As Integer
    ' integer untuk menyimpan nilai

    iHasilPencarian = 0

    ' Loop di kolom A hingga ditemukan.

    For i = 1 To 100
        If Cells(i, 1).Value = sTeksPencarian Then

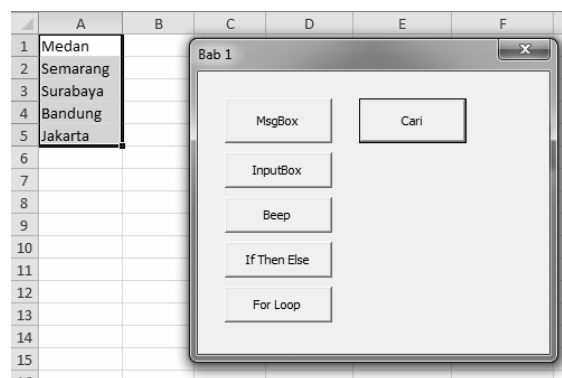
            ' kalau ditemukan baris kolom akan dicatat
            iHasilPencarian = i
            Exit For

        End If
    Next i

    ' Menampilkan message box
    If iHasilPencarian = 0 Then
        MsgBox "String " & sTeksPencarian & " tidak ditemukan"
    Else
        MsgBox "String " & sTeksPencarian & " ditemukan di sel
A" & iHasilPencarian
    End If

End Sub
```

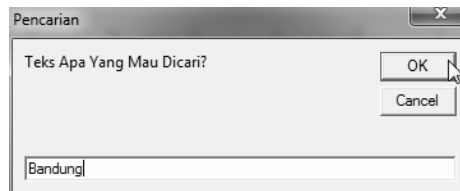
4. Sekarang jalankan program. Dan ketikkan beberapa teks di kolom A.



**Gambar 1.40** Kolom-kolom di A berisi teks

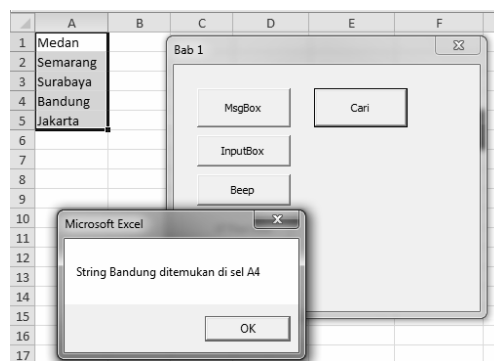


5. Kemudian klik tombol Cari, muncul input box, isikan teks yang akan dicari, misalnya kota “Bandung”.



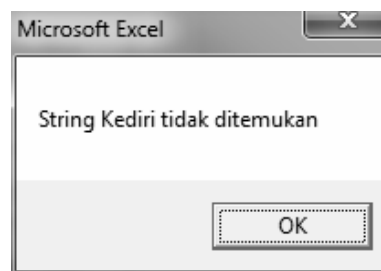
**Gambar 1.41 Pengisian teks yang akan dicari**

6. Maka kalau teks yang dicari ada di kolom A, maka ada messagebox yang menjelaskan lokasi string tersebut.



**Gambar 1.42 Lokasi string ditemukan di kolom A**

7. Kalau tidak ada yang ditemukan, muncul messagebox yang menjelaskan bahwa string yang dicari tidak ditemukan.



**Gambar 1.43 Messagebox yang menyatakan string tidak ditemukan**

### 1.10.3 Statement Select Case

Seperti telah dijelaskan sebelumnya, konstruk If Then digunakan untuk mengatur flow program berdasarkan berbagai kemungkinan. Begitu pula Select Case ini, fungsinya kurang lebih sama, hanya saja cocok digunakan kalau fitur kemungkinannya lebih banyak.

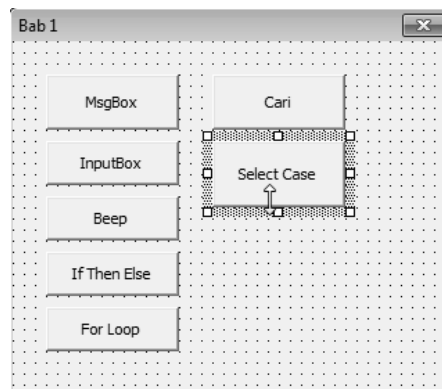
Sintaks dari statement Select Case ini adalah:

```
Select Case EkspresiTes
  Case value1
    ' statement yang akan dieksekusi jika EkspresiTes = value1
  Case value2
    ' statement yang akan dieksekusi jika EkspresiTes = value2
  . . .

  Case Else
    ' statement yang akan dieksekusi jika tidak ada yang sesuai
    dengan case yang sudah disebutkan sebelumnya.
End Select
```

Bagian Case Else sifatnya opsional, jadi Anda tidak wajib untuk menyertakannya. Berikut ini contoh penggunaan Select Case untuk Excel VBA:

1. Masukkan tombol baru di form, dan ganti namanya dengan **Select Case**.



*Gambar 1.44 Memasukkan button dengan Select case*

2. Klik 2x pada tombol **Select Case**.
3. Isikan kode berikut:

```

Private Sub CommandButton7_Click()
' Mengecek apakah teks antara huruf A dan E
  Select Case Range("C2").Text

    Case "A" To "Ezzzzzz"

      Range("C3").Value = "Ada di antaranya"

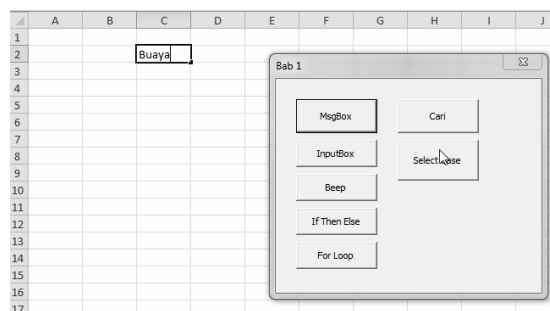
    Case Else

      Range("C3").Value = "Tidak ada diantaranya"

  End Select
End Sub

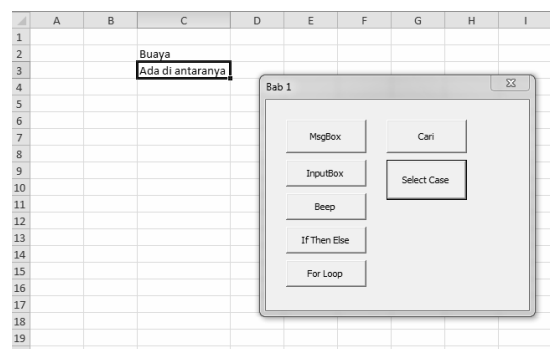
```

4. Apabila dijalankan, kemudian di sel C2, Anda menuliskan kata yang hurufnya antara huruf A dan E (misalnya Buaya), lalu klik **Select Case**.



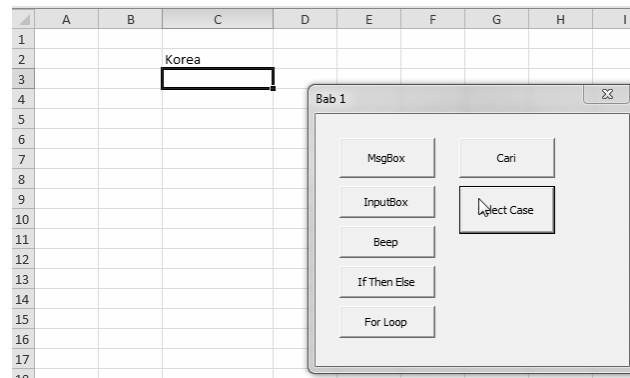
**Gambar 1.45 Klik pada Select Case**

5. Maka di sel C3, muncul tulisan bahwa huruf tersebut ada di antara A dan E.



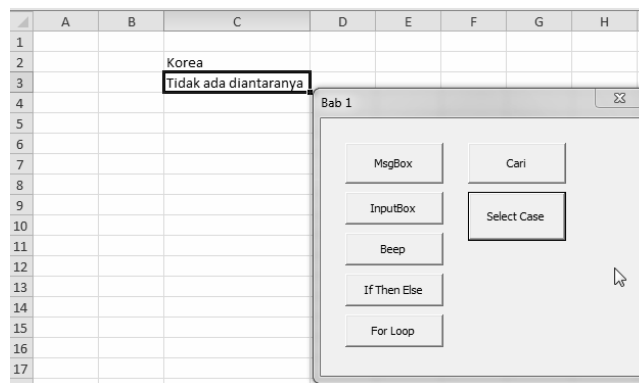
**Gambar 1.46 Huruf ada di antara A dan E**

6. Sekarang coba ketikkan huruf di C2 yang tidak ada di antara A dan E.



*Gambar 1.47 Pengetikan huruf yang tidak ada di antara A dan E*

7. Kalau diklik Select Case, ada tulisan **Tidak ada di antaranya**.



*Gambar 1.48 Ada tulisan “Tidak ada di antaranya”*

## 1.11 Objek dan Fungsi Excel

Untuk melakukan pemrograman dengan optimal di Excel menggunakan VBA, Anda harus lebih mengenal objek dan fungsi Excel. Subbab berikut ini menjelaskan beberapa objek dan fungsi Excel.

### 1.11.1 Fungsi-Fungsi yang Berkaitan dengan File

VBA memiliki banyak fungsi yang berkaitan dengan file dan direktori. Berikut ini beberapa fungsi penting yang perlu Anda ketahui untuk melakukan pemrograman berkaitan dengan penanganan file dan direktori.

Fungsi	Deskripsi
Dir	Mencari file dengan nama tertentu.
FileLen	Memperoleh panjang dari file.
FileTimeDate	Mengambil tanggal stamp dari file.
FileCopy	Menyalin file.
Kill	Menghapus file.
Name	Rename file atau direktori.
RmDir	Menghapus direktori.
MkDir	Membuat direktori baru.

### 1.11.2 Event Worksheet

Model objek dari Excel memiliki beberapa objek, seperti AppEvents, DocEvents, ChartEvents, WorkbookEvents, OLEObjectEvents, dan RefreshEvents. Event-event yang diasosiasikan dengan worksheet misalnya, adalah method dari objek DocEvents yang merupakan anakan dari objek Worksheet dan objek Chart.

Berikut ini event-event yang berkaitan dengan worksheet. Event ini juga dianggap sebagai event dokumen.

Nama Event	Deskripsi
Activate	Terjadi ketika worksheet diaktifkan.
BeforeDoubleClick	Terjadi ketika worksheet diklik ganda, dan tepat sebelumnya event ini akan dipicu.
BeforeRightClick	Terjadi ketika worksheet diklik kanan, sebelum action default dari klik kanan.

Calculate	Terjadi setelah worksheet dikalkulasi ulang.
Change	Terjadi ketika sel di worksheet diubah oleh user menggunakan link eksternal.
Deactivate	Terjadi ketika worksheet dinonaktifkan.
PivotTableUpdate	Terjadi ketika laporan PivotTable di-update pada worksheet.
SelectionChange	Terjadi ketika ada perubahan seleksi di worksheet.

### 1.11.3 Event Workbook

Event workbook berkaitan erat dengan file Excel secara keseluruhan (workbook). Beberapa event penting di workbook ini dapat dilihat di tabel berikut.

Nama Event	Deskripsi
Activate	Terjadi ketika workbook diaktifkan.
AddinInstall	Terjadi ketika workbook diinstal sebagai add in.
AddinUninstall	Terjadi ketika workbook di-uninstall sebagai add-in.
BeforeClose	Terjadi sebelum workbook ditutup.
BeforePrint	Terjadi sebelum workbook dicetak.
BeforeSave	Terjadi sebelum workbook disimpan.
Deactivate	Terjadi ketika workbook dinonaktifkan.
NewSheet	Terjadi ketika sheet baru dibuat di workbook.
Open	Terjadi ketika workbook dibuka.
PivotTableCloseConnection (Excel 10 only)	Terjadi setelah PivotTable menutup koneksi ke data source.

PivotTableOpenConnection(Excel 10 only)	Terjadi setelah PivotTable membuka koneksi ke data source-nya.
SheetActivate	Terjadi ketika ada sheet yang diaktifkan.
SheetBeforeDoubleClick	Terjadi ketika ada worksheet diklik 2x, ini terjadi sebelum klik 2x terjadi.
SheetBeforeRightClick	Terjadi ketika ada klik kanan di worksheet, terjadi tepat sebelum adanya klik kanan.
SheetCalculate	Terjadi setelah ada worksheet direkalkulasi. Atau ada data yang diubah sudah diplot di chart.
SheetChange	Terjadi ketika sel di worksheet diubah oleh user atau oleh link eksternal.
SheetDeactivate	Terjadi ketika ada sheet yang dinonaktifkan.
SheetSelectionChange	Terjadi ketika ada perubahan seleksi di worksheet. Tidak akan terjadi jika seleksi di sheet chart.
WindowActivate	Terjadi hanya ketika jendela workbook diaktivasi.
WindowDeactivate	Terjadi ketika ada workbook dinonaktifkan.
WindowResize	Terjadi ketika jendela workbook di-resize.

#### 1.11.4 Event Chart

Chart adalah grafik yang lazim ada di worksheet. Anda bisa melihat beberapa event dari chart dengan melihat tabel berikut ini.

Nama Event	Deskripsi
Activate	Terjadi ketika sheet chart atau chart yang di-embedded diaktifkan.

BeforeDoubleClick	Terjadi ketika embedded chart diklik 2x, ini terjadi tepat sebelum klik 2x tersebut.
BeforeRightClick	Terjadi ketika chart embedded diklik kanan, ini terjadi tepat sebelum klik kanan terjadi.
Calculate	Terjadi setelah chart memplot data baru atau data yang berubah.
Deactivate	Terjadi ketika chart dinonaktifkan.
DragOver	Terjadi ketika range sel di-drag di atas chart.
DragPlot	Terjadi ketika range sel di-drag dan di-drop ke chart.
MouseDown	Terjadi ketika tombol mouse diklik sementara pointer sedang ada di atas chart.
MouseMove	Terjadi ketika posisi pointer mouse berubah di atas chart.
MouseUp	Terjadi ketika tombol mouse dilepas, sementara pointer ada di atas chart.
Resize	Terjadi ketika chart di-resize.
Select	Terjadi ketika elemen chart dipilih.
SeriesChange	Terjadi ketika user mengubah nilai dari poin data di chart.